WHAT IS CLAIMED IS:

1. A method for controlling access to an application by a client or user, the method comprising:

a) if identification information is not mapped to an activation key associated with the application,

i) identifying the activation key to be sent to the client, based on the application; and

ii) mapping the activation key to the identification information; and

b) sending the activation key to the client.

2. The method of claim 1, further comprising:

c) delivering content to the client, wherein the content enables the client to execute the application.

3. The method of claim 1, further comprising:

c) determining whether the application requires an activation key,

performed before step a).

4. The method of claim 1, further comprising:

a) iii) determining whether an activation key is available,

performed before step a) i).

5. The method of claim 1, wherein the identification information represents the identity of the user.

6. The method of claim 1, wherein the identification information represents the identity of the client.

7. The method of claim 1, further comprising:

c) encoding the activation key,
performed before step b).

8. The method of claim 1, further comprising:

c) placing the activation key on the client computer in a place and format expected by the application,
performed after step b).

9. The method of claim 1, further comprising:

c) encoding the activation key according to a format expected by the application wherein said encoding is performed in a manner specific to the client,
performed before step b).

10. A method of utilizing an activation key to indicate authorization to use an application, the method comprising the steps of:

a) receiving the activation key, from a vendor server, at a client;

b) storing the activation key locally to the client in a manner determined by the application;

c) executing the application to perform security processing, to determine whether continued use of the application is permitted.

11. The method of claim 8, further comprising:

d) encoding the activation key,
performed after step a) and before step b).

12. A system for controlling access to an application by a prospective user, comprising:

a database that maps an activation key to identification information; and

a vendor server that receives said activation key from said database and sends said activation key to the user.

13. The system of claim 10, wherein said identification information represents the identity of the user.

14. The system of claim 10, wherein said identification information represents the identity of the client.

15. The system of claim 10, wherein said activation key is one of a plurality of activation keys stored at said database and associated with the application.

16. The system of claim 13, wherein for each of a plurality of applications, said database stores a respective plurality of activation keys.

17. A system for managing the use of application licenses in an organization , comprising:

(a) a database of license keys in a central server;

(b) a client in communication with said database; and

(c) an instance of the application on said client, wherein each instance of the application that is executed on the client gets an unallocated activation key from said central database and registers the key as allocated

18. A method of upgrading content of an online delivered application, comprising the steps of:

(a) creating an upgraded image of the application at an application server, based on the upgraded content of the application;

(b) informing the client of the identity of one or more blocks of the upgraded image that have been changed; and

(c) delivering any changed blocks requested by the client.

19. A method of upgrading content of an online delivered application at a client, comprising the steps of:

(a) copying a file with one or more locally updated blocks from a dynamic cache to a static cache;

(b) deleting the file with one or more locally updated blocks from the dynamic cache;

(c) copying the file with one or more locally updated blocks and any newly locally created files from the static cache to a backup directory;

(d) clearing the static cache;

(e) receiving the identity of one or more blocks of an upgraded image;

(f) for any current block, held in the dynamic cache, that corresponds to an identified upgraded block, deleting the corresponding current block from the primary cache;

(g) loading the locally updated blocks and the newly created blocks from the backup directory to the static cache; and

(h) downloading the identified upgraded blocks to the dynamic cache as necessary.

20. The method of claim 19, wherein the dynamic cache address of any block in the dynamic cache is determined by hashing the block, such that the hash result corresponds to the dynamic cache address for the block.

21. A system for upgrading client content of an online delivered application, comprising:

a dynamic cache at the client that stores one or more current blocks of an image of the application;

a static cache at the client that receives any locally updated files and any locally created new files; and

a backup directory that receives said locally updated files and said new files from said static cache,

wherein said dynamic cache receives upgraded blocks of an upgraded image from an application server, said static cache is cleared after said backup directory receives said locally updated files and said new files from said static cache, and said static cache receives said locally updated files and said new files from said backup directory.

22. The system of claim 21, further comprising:

logic for hashing the contents of each block to be stored in said dynamic cache, to produce a respective hash value that corresponds to a dynamic cache address for the respective block.

23. The system of claim 21, further comprising:

logic for hashing the contents of each blocks to be stored in said static cache, to produce a respective hash value that corresponds to a static cache address for the respective block.

24. A system for overlaying information on an application display, comprising:

logic for device creation, such that when said device creation logic is executed, access to an application program interface is retained after said logic for device creation has completed execution;

logic for obtaining a process address, such that execution of said logic for obtaining a process address returns an address corresponding to said logic for device creation; and

logic for a library loading, such that said logic for library loading includes an address corresponding to said logic for a device creation.

25. The system of claim 24, wherein said application programming interface is a version of DirectX.

26. The system of claim 24, wherein said application programming interface is a version of OpenGL.

27. A method of overlaying information on an application display, comprising:

(a) initializing an information object;

(b) substituting device creation logic for previous device creation logic, wherein said device creation logic retains access to an application programming interface after said device creation logic has completed executing;

(c) substituting logic for obtaining a process address for previous logic for obtaining a process address, wherein said logic for obtaining a process address retains an address corresponding to said device creation logic;

(d) substituting library loading logic for previous library loading logic, wherein said library loading logic includes an address corresponding to said device creation logic; and

(e) rendering said information using a device created and stored by said device creation logic.

28. The method of claim 27, wherein said application programming interface is a version of DirectX.

29. The method of claim 27, wherein said application programming interface is a version of OpenGL.